

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$

synthesis is realized by composing rule consequents with variable weights. This idea conforms to the principle of variable weighted synthesis in multifactorial decision-making.

The notion of variable weights was introduced for the synthetic decision-making analysis in [8]. Reference [9] gave the axiomatic definitions of variable weight vectors and state variable weight vectors in accordance with varying regularity of weights, and presented the principle of variable weights saying that “variable weight vectors are made by normalized Hadamard products of a constant weight vector and state variable weight vectors.” The concept of synthesis functions, also named multifactorial functions, was introduced for reducing dimensions in [10]. If the parameters in synthesis functions are variable, the synthesis functions are called variable weighted synthesis functions and have the ability of merging information.

In this paper, we propose the variable weighted synthesis inference (VWSI) method by applying the principle of variable weighted synthesis to fuzzy inference. In Section 2, we give some preliminary knowledge. In Section 3, we introduce the notions of states of rules, state variable weights of rules and variable weights of rules, and also present the VWSI method and some algorithmic models. In Section 4, we discuss the construction of variable weights of rules. In Section 5, we analyze the interpolation mechanism of fuzzy systems constructed by several VWSI algorithmic models. In Section 6, we discuss the relationship between the fuzzy systems based on VWSI algorithms and the fuzzy systems based on commonly used fuzzy inference algorithms. In Section 7, we give a method for determining constant weights of rules and a simulation experiment.

2. PRELIMINARIES

DEFINITION 2.1. *An m -dimensional variable weight is a mapping $W = (w_1, \dots, w_m)$ from $[0, 1]^m$ to $[0, 1]^m$, where*

$$w_i : [0, 1]^m \rightarrow [0, 1]; \quad (x_1, \dots, x_m) \mapsto w_i(x_1, \dots, x_m), \quad 1 \leq i \leq m.$$

A variable weight with reward is an m -dimensional variable weight W satisfying the following three conditions:

- (w.1) *normality, i.e., $\sum_{i=1}^m w_i(x_1, \dots, x_m) = 1$,*
- (w.2) *continuity, i.e., every mapping w_i is continuous with respect to any variable x_j ($1 \leq i, j \leq m$),*
- (w.3) *reward law, i.e., every mapping w_i is monotonically increasing with respect to the variable x_i ($1 \leq i \leq m$).*

A variable weight with penalty is an m -dimensional variable weight W satisfying (w.1), (w.2) and the following condition:

- (w.3') *penalty law, i.e., every mapping w_i is monotonically decreasing with respect to the variable x_i ($1 \leq i \leq m$).*

A synthesis function M_m , also named a multifactorial function, is essentially a projection from m -dimensional space to one-dimensional space [10]. In many cases, the spaces may be transformed into the unit closed intervals. Then $M_m : [0, 1]^m \rightarrow [0, 1]$ and it is called a standard synthesis function. Standard synthesis functions can be divided into additive ones and nonadditive ones.

DEFINITION 2.2. *An additive m -ary standard synthesis function is a mapping $M_m : [0, 1]^m \rightarrow [0, 1]; (x_1, \dots, x_m) \mapsto M_m(x_1, \dots, x_m)$ satisfying the following three conditions:*

- (m.1) $x_i \leq y_i$ ($1 \leq i \leq m$) $\Rightarrow M_m(x_1, \dots, x_m) \leq M_m(y_1, \dots, y_m)$,
- (m.2) M_m is continuous with respect to any variable x_i ($1 \leq i \leq m$),
- (m.3) $\bigwedge_{i=1}^m x_i \leq M_m(x_1, \dots, x_m) \leq \bigvee_{i=1}^m x_i$.

A nonadditive m -ary standard synthesis function is a mapping

$$M_m : [0, 1]^m \rightarrow [0, 1]; \quad (x_1, \dots, x_m) \mapsto M_m(x_1, \dots, x_m),$$

satisfying (m.1), (m.2) and the following condition:

$$(m.3') \quad M_m(x_1, \dots, x_m) > \bigvee_{i=1}^m x_i \text{ or } M_m(x_1, \dots, x_m) < \bigwedge_{i=1}^m x_i.$$

EXAMPLE 2.1. The following mappings from $[0, 1]^m$ to $[0, 1]$ are additive m -ary standard synthesis functions:

$$\bigwedge(x_1, \dots, x_m) \triangleq \bigwedge_{i=1}^m x_i; \quad (1)$$

$$\bigvee(x_1, \dots, x_m) \triangleq \bigvee_{i=1}^m x_i; \quad (2)$$

$$\sum(x_1, \dots, x_m) \triangleq \sum_{i=1}^m w_i x_i, \quad (3)$$

where $w_i \in [0, 1]$ and $\sum_{i=1}^m w_i = 1$;

$$M_m(x_1, \dots, x_m) \triangleq \bigvee_{i=1}^m w_i x_i, \quad (4)$$

where $w_i \in [0, 1]$ and $\bigvee_{i=1}^m w_i = 1$;

$$M_m(x_1, \dots, x_m) \triangleq \bigvee_{i=1}^m (w_i \wedge x_i), \quad (5)$$

where $w_i \in [0, 1]$ and $\bigvee_{i=1}^m w_i = 1$;

$$M_m(x_1, \dots, x_m) \triangleq \left(\sum_{i=1}^m w_i x_i^p \right)^{1/p}, \quad (6)$$

where $p > 0$, $w_i \in [0, 1]$ and $\sum_{i=1}^m w_i = 1$.

EXAMPLE 2.2. The mapping $\prod : [0, 1]^m \rightarrow [0, 1]$ given by

$$\prod(x_1, \dots, x_m) \triangleq \prod_{i=1}^m x_i \quad (7)$$

is a nonadditive m -ary standard synthesis function.

NOTE 2.1. In (3)–(6), if every parameter w_i is variable, that is, $w_i \triangleq w_i(x_1, \dots, x_m)$ ($1 \leq i \leq m$), then those synthesis functions are said to be variable weighted.

In [9], it was pointed out that in variable weighted synthesis, the relative importance of every factor should be considered and the weights should vary with different factor states.

DEFINITION 2.3. Let $S = (S_1, \dots, S_m)$ be a mapping from $[0, 1]^m$ to $[0, 1]^m$, where

$$S_i : [0, 1]^m \rightarrow [0, 1]; \quad (x_1, \dots, x_m) \mapsto S_i(x_1, \dots, x_m), \quad 1 \leq i \leq m.$$

The mapping S is called an m -dimensional state variable weight with reward if it satisfies the following four conditions:

(s.1) S is symmetric, i.e., for any i, j ($1 \leq i < j \leq m$),

$$S(x_1, \dots, x_i, \dots, x_j, \dots, x_m) = S(x_1, \dots, x_j, \dots, x_i, \dots, x_m),$$

(s.2) every $S_i(x_1, \dots, x_m)$ is continuous with respect to any variable x_j ($1 \leq i, j \leq m$),

(s.3) $x_i \geq x_j \Rightarrow S_i(x_1, \dots, x_m) \geq S_j(x_1, \dots, x_m)$,

(s.4) the mapping $W : [0, 1]^m \rightarrow [0, 1]^m$ given by

$$X \mapsto W(X) \triangleq \frac{W^0 \circ S(X)}{\sum_{i=1}^m w_i^0 S_i(X)}$$

is a variable weight with reward, where $W^0 = (w_1^0, \dots, w_m^0)$ is a constant weight vector and “ \circ ” is the Hadamard product operator.

The mapping S is called an m -dimensional state variable weight with penalty if it satisfies (s.1), (s.2) and the following two conditions:

(s.3') $x_i \geq x_j \Rightarrow S_i(x_1, \dots, x_m) \leq S_j(x_1, \dots, x_m)$,

(s.4') the mapping $W : [0, 1]^m \rightarrow [0, 1]^m$ given by

$$X \mapsto W(X) \triangleq \frac{W^0 \circ S(X)}{\sum_{i=1}^m w_i^0 S_i(X)}$$

is a variable weight with penalty, where $W^0 = (w_1^0, \dots, w_m^0)$ is a constant weight vector and “ \circ ” is the Hadamard product operator.

The principle of variable weights, presented in [9], is as follows.

THEOREM 2.1. *The variable weight vector $W(X)$ is the normalized Hadamard product of a constant weight vector W^0 and a state variable weight vector $S(X)$, i.e.,*

$$W(X) = \frac{W^0 \circ S(X)}{\sum_{i=1}^m w_i^0 \cdot S_i(X)}.$$

3. VWSI METHOD FOR FUZZY REASONING

We herein primarily discuss the multiple fuzzy inference, since it is widely applied in practice. The general form of multiple fuzzy inference is given as follows:

$$\begin{array}{ll} \text{Known} & \text{If } x \text{ is } A_1 \text{ then } y \text{ is } B_1 \\ & \text{If } x \text{ is } A_2 \text{ then } y \text{ is } B_2 \\ & \dots \\ & \text{If } x \text{ is } A_m \text{ then } y \text{ is } B_m \\ \text{Given} & x \text{ is } A' \\ \text{Determined} & y \text{ is } B' \end{array}$$

where $x \in X$, $y \in Y$, $A_i, A' \in \mathcal{F}(X)$ and $B_i, B' \in \mathcal{F}(Y)$ ($1 \leq i \leq m$).

We first intend to show the principle of variable weighted synthesis in fuzzy inference through analyzing the inference process of the CRI method on the basis of the Mamdani implication operator. Every inference rule is regarded as a fuzzy relation R_i from X to Y defined by the Mamdani implication “ \wedge ”, i.e., $R_i(x, y) = A_i(x) \wedge B_i(y)$ ($1 \leq i \leq m$), and m inference rules are jointed by fuzzy set operator “ \vee ” to get the total inference relation $R(x, y)$, that is,

$$R(x, y) \triangleq \bigvee_{i=1}^m R_i(x, y) = \bigvee_{i=1}^m (A_i(x) \wedge B_i(y)).$$

Then the given input fuzzy set A' and the total inference relation $R(x, y)$ are composed by fuzzy relation composite operation to obtain the output fuzzy set B' , i.e.,

$$B'(y) = \bigvee_{x \in X} \left[A'(x) \wedge \bigvee_{i=1}^m (A_i(x) \wedge B_i(y)) \right] = \bigvee_{i=1}^m \left[\left(\bigvee_{x \in X} (A'(x) \wedge A_i(x)) \right) \wedge B_i(y) \right]. \quad (8)$$

From another viewpoint, equation (8) can be explained as follows: $\bigvee_{x \in X} (A'(x) \wedge A_i(x))$ is the approach degree between input fuzzy set A' and rule antecedent A_i (this formula satisfies the axioms of approach degree of fuzzy sets, refer to [11]). The value of approach degree can be regarded as a variable weight of the i^{th} rule in the inference process, since it changes with A' . And $(\bigvee_{x \in X} (A'(x) \wedge A_i(x))) \wedge B_i(y)$ shows that the rule consequent B_i is weighted by $\bigvee_{x \in X} (A'(x) \wedge A_i(x))$. Finally, m rule consequents with variable weights are synthesized by disjunctive operator “ \vee ” to obtain the output fuzzy set B' . In conclusion, the whole inference process conforms to the principle of variable weighted synthesis in the multifactorial decision-making.

The principle of variable weighted synthesis can be taken to realize fuzzy inference. In fact, it is a reasonable inference process for simulating man's inference thought that first the weight of every inference rule is determined by some fuzzy relation between the given input fuzzy set and every rule antecedent, and then rule consequents with the weights are synthesized to produce the output fuzzy set. Accordingly, we propose a new idea of fuzzy inference, that is, an idea on variable weighted synthesis of fuzzy inference.

Some necessary definitions should be introduced. We think that inference rules in the rule-base are static, in other words, their initial states are zero before making inference. When the given input fuzzy set A' is compared with every rule antecedent in the rule-base, a state value is assigned to each rule.

DEFINITION 3.1. Let s be a fuzzy relation from $\mathcal{F}(X)$ to $\mathcal{F}(X)$. Then $s(A', A_i)$ is called the state of the i^{th} rule for the input fuzzy set A' .

DEFINITION 3.2. Let $\mathcal{D}_s = \{(s(A', A_1), \dots, s(A', A_m)) \mid A' \in \mathcal{F}(X)\}$. An m -dimensional state variable weight with reward of rules is a mapping $S = (S_1, \dots, S_m) : \mathcal{D}_s \rightarrow [0, 1]^m$ satisfying the following five conditions:

- (S.1) normality, i.e., $\sum_{i=1}^m S_i(s(A', A_1), \dots, s(A', A_m)) = 1$,
- (S.2) continuity, i.e., every S_i is continuous with respect to any $s(A', A_j)$ ($1 \leq i, j \leq m$),
- (S.3) order-preserving, i.e.,

$$s(A', A_i) \geq s(A', A_j) \Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) \geq S_j(s(A', A_1), \dots, s(A', A_m)),$$

- (S.4) reward law, i.e., S_i is monotonically increasing with respect to $s(A', A_i)$ ($1 \leq i \leq m$),
- (S.5) polarity, i.e.,

$$\begin{aligned} s(A', A_i) = 0 &\Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) = 0, \\ s(A', A_i) = 1 &\Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) = 1. \end{aligned}$$

DEFINITION 3.3. Let $\mathcal{D}_s = \{(s(A', A_1), \dots, s(A', A_m)) \mid A' \in \mathcal{F}(X)\}$. An m -dimensional state variable weight with penalty of rules is a mapping $S = (S_1, \dots, S_m) : \mathcal{D}_s \rightarrow [0, 1]^m$ satisfying (S.1), (S.2) and the following three conditions:

- (S.3') order-inverting, i.e.,

$$s(A', A_i) \geq s(A', A_j) \Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) \leq S_j(s(A', A_1), \dots, s(A', A_m)),$$

- (S.4') penalty law, i.e., S_i is monotonically decreasing with respect to $s(A', A_i)$ ($1 \leq i \leq m$),
- (S.5') polarity, i.e.,

$$\begin{aligned} s(A', A_i) = 0 &\Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) = 1, \\ s(A', A_i) = 1 &\Rightarrow S_i(s(A', A_1), \dots, s(A', A_m)) = 0. \end{aligned}$$

DEFINITION 3.4. Let $W^0 = (w_1^0, \dots, w_m^0)$ be a constant weight vector of rules, and $S = (S_1, \dots, S_m)$ be a state variable weight with reward (with penalty) of rules. A variable weight with reward (with penalty) of rules is a mapping $W : \mathcal{F}(X) \rightarrow [0, 1]^m$ defined by

$$A' \mapsto W(A') = \frac{W^0 \circ S(s(A', A_1), \dots, s(A', A_m))}{\sum_{i=1}^m w_i^0 \cdot S_i(s(A', A_1), \dots, s(A', A_m))}.$$

NOTE 3.1. In Definition 3.4, the constant weight vector W^0 of rules expresses the credibility of rules, since those rules which are extracted from input-output data or summarized by area experts are not always absolutely credible. In Section 7, we will give an approach of determining W^0 by using known input-output data and present a simulated experiment.

On the basis of the above results, we introduce the VWSI (variable weighted synthesis inference) method in a step-by-step manner.

STEP 1. By input-output data or area experts' experience, a constant weight vector $W^0 = (w_1^0, \dots, w_m^0)$ is determined to represent the relative credibility of rules.

STEP 2. A state variable weight $S_i(s(A', A_1), \dots, s(A', A_m))$ of every rule is determined by some fuzzy relation s between input fuzzy set A' and rule antecedent A_i ($1 \leq i \leq m$).

STEP 3. According to m state variable weights of rules, choose a variable weight vector $W(A') = (w_1(A'), \dots, w_m(A'))$ of rules for A' as the activation degrees of rules.

STEP 4. The output fuzzy set is obtained from rule consequents by the variable weighted synthesis.

According to this method, we need to choose the form of concrete synthesis functions. The following give several algorithmic models of variable weighted synthesis for determining the output fuzzy set $B' \in \mathcal{F}(Y)$.

MODEL 1.

$$B'(y) = \sum_{i=1}^m w_i(A') B_i(y). \quad (9)$$

MODEL 2.

$$B'(y) = \bigvee_{i=1}^m w_i(A') B_i(y). \quad (10)$$

MODEL 3.

$$B'(y) = \bigvee_{i=1}^m (w_i(A') \wedge B_i(y)). \quad (11)$$

MODEL 4.

$$B'(y) = \left(\sum_{i=1}^m w_i(A') B_i^p(y) \right)^{1/p}, \quad p > 0. \quad (12)$$

NOTE 3.2. According to Definition 3.4, a variable weight of every rule consists of its constant weight and state variable weight. Constant weights of rules are determined before making inference. Only state variable weights of rules are relevant to the given input fuzzy set A' , and they have the function of adjusting weights assigned to rules in accordance with A' . Therefore, the determination of state variable weights of rules seems critical in fuzzy inference.

4. CONSTRUCTION OF VARIABLE WEIGHTS OF RULES

4.1. Variable Weight with Reward of Rules Based on Approach Degree of Fuzzy Sets

The definition of approach degree of fuzzy sets was introduced in [12].

DEFINITION 4.1.1. Let $\sigma : \mathcal{F}(X) \times \mathcal{F}(X) \rightarrow [0, 1]$; $(A, B) \mapsto \sigma(A, B)$ be a mapping satisfying the following four conditions:

- ($\sigma.1$) $\sigma(A, A) = 1$,
- ($\sigma.2$) $\sigma(\phi, X) = 0$,
- ($\sigma.3$) $\sigma(A, B) = \sigma(B, A)$,
- ($\sigma.4$) $A \subseteq B \subseteq C \Rightarrow \sigma(A, C) \leq \sigma(A, B) \wedge \sigma(B, C)$.

Then $\sigma(A, B)$ is called an approach degree of fuzzy sets A and B .

EXAMPLE 4.1.1. Let $\sigma(A, B) = \bigvee_{x \in X} (A(x) \wedge B(x))$ for fuzzy sets A and B . Then σ satisfies four conditions in Definition 4.1.1. Put $s(A', A_i) \triangleq \sigma(A', A_i)$,

$$S_i(\sigma(A', A_1), \dots, \sigma(A', A_m)) \triangleq \sigma(A', A_i)$$

and

$$w_i(A') \triangleq \frac{w_i^0 \cdot S_i(\sigma(A', A_1), \dots, \sigma(A', A_m))}{\sum_{j=1}^m w_j^0 \cdot S_j(\sigma(A', A_1), \dots, \sigma(A', A_m))}. \quad (13)$$

Then $w_i(A')$ is a variable weight with reward of the i^{th} rule, where w_i^0 is a constant weight of the i^{th} rule ($1 \leq i \leq m$).

4.2. Variable Weight with Penalty of Rules Based on Distance of Fuzzy Sets

There are many kinds of distances between fuzzy sets. Considering the special structure of fuzzy systems in practice, we here use the center of peak points of a fuzzy set, namely, the center of $\{x \in X \mid A(x) = 1\}$. Let

$$d(A, B) = \alpha |x' - x|,$$

where α is a normalized factor, and x' and x are centers of peak points of fuzzy sets A and B , respectively. Put $s(A', A_i) \triangleq d(A', A_i)$, $S_i(d(A', A_1), \dots, d(A', A_m)) \triangleq 1 - d(A', A_i)$ and

$$w_i(A') \triangleq \frac{w_i^0 \cdot S_i(d(A', A_1), \dots, d(A', A_m))}{\sum_{j=1}^m w_j^0 \cdot S_j(d(A', A_1), \dots, d(A', A_m))}. \quad (14)$$

Then $w_i(A')$ is a variable weight with penalty of the i^{th} rule, where w_i^0 is a constant weight of the i^{th} rule ($1 \leq i \leq m$).

4.3. Variable Weight of Rules Based on Fuzzy Implication Operators

In the CRI method for fuzzy inference, the selection of fuzzy implication operators is a key of inference and has great influence on inference results. In the VWSI method, we can also use fuzzy implication operators to construct variable weights of rules. The axiomatic definition of fuzzy implication operators was given in [13].

DEFINITION 4.3.1. A fuzzy implication operator is a mapping $\theta : [0, 1]^2 \rightarrow [0, 1]$ satisfying the following two conditions:

- (I.1) there exists $a, b \in [0, 1]$ such that $\theta(a, b) = 1$,
- (I.2) there exists $a', b' \in [0, 1]$ such that $\theta(a', b') = 0$.

A normal implication operator is a fuzzy implication operator θ satisfying

- (I.3) $\theta(0, 0) = \theta(0, 1) = \theta(1, 1) = 1$, $\theta(1, 0) = 0$.

Obviously, normal implication operators are a generalization of two-valued logical implication operator.

In the CRI method, fuzzy relations between fuzzy sets are represented by fuzzy implication operators. Although many fuzzy implication operators such as Mamdani implication, Larsen product implication, Bounded-product implication, etc., are not normal, the fuzzy systems constructed by them have good properties [14]. Meanwhile, a lot of normal implication operators such as Lukasiewicz implication, Goguen implication, Gödel implication, and so on, have no significance for construction of practical control systems [15]. For the sake of generalization of two-valued logic, we choose normal implication operators to construct variable weights of rules.

EXAMPLE 4.3.1. Suppose θ is a normal implication operator. We define $s(A', A_i) \triangleq \theta(A', A_i)$, $S_i(\theta(A', A_1), \dots, \theta(A', A_m)) \triangleq (\bigwedge_{x \in X} \theta(A'(x), A_i(x))) \vee (\bigwedge_{x \in X} \theta(A_i(x), A'(x)))$ and

$$w_i(A') \triangleq \frac{w_i^0 \cdot S_i(\theta(A', A_1), \dots, \theta(A', A_m))}{\sum_{j=1}^m w_j^0 \cdot S_j(\theta(A', A_1), \dots, \theta(A', A_m))}, \quad (15)$$

where w_i^0 is a constant weight of the i^{th} rule ($1 \leq i \leq m$).

NOTE 4.3.1. $\theta(A', A_i)$ represents a fuzzy relation between A' and A_i , and different θ reflects distinct relevance between A' and A_i . So $\theta(A', A_i)$ can express the state of the i^{th} rule.

NOTE 4.3.2. $w_i(A')$ defined by (15) is neither a variable weight with reward of the i^{th} rule nor a variable weight with penalty of the i^{th} rule. However, we can prove that the fuzzy system constructed by $w_i(A')$ has a good interpolation approximation characteristic.

5. INTERPOLATION MECHANISM OF FUZZY SYSTEMS BASED ON VWSI ALGORITHMIC MODELS

The universal approximation of fuzzy systems is a theoretical foundation of their applications in practice. In this section, we analyze the interpolation approximation ability of fuzzy systems based on several VWSI algorithmic models from the analytical viewpoint. For convenience, single-input single-output fuzzy systems are discussed.

A fuzzy partition of a set X is a family $\mathcal{A} = \{A_i \mid 1 \leq i \leq m\}$ of normal fuzzy sets on X satisfying the following conditions: $\sum_{i=1}^m A_i(x) \equiv 1$. Then A_i ($1 \leq i \leq m$) are called base elements of \mathcal{A} , and \mathcal{A} is also called a group of base elements of X .

Let X be the universe of input variable and Y be the universe of output variable. Denote by $\mathcal{A} \triangleq \{A_i \mid 1 \leq i \leq m\}$ and $\mathcal{B} \triangleq \{B_i \mid 1 \leq i \leq m\}$ fuzzy partitions of X and Y , respectively. Without the loss of generality, we can suppose $X = [a, b]$ and $Y = [c, d]$ are real number intervals, and the following hold:

$$a \leq x_1 < x_2 < \dots < x_m \leq b, \quad c \leq y_1 < y_2 < \dots < y_m \leq d,$$

where x_i and y_i represent peak points of A_i and B_i ($1 \leq i \leq m$), respectively. If we regard \mathcal{A} and \mathcal{B} as linguistic variables, then m fuzzy inference rules can be formed as follows:

$$\text{If } x \text{ is } A_i \text{ then } y \text{ is } B_i \text{ } (1 \leq i \leq m).$$

For a practical fuzzy system, its input is a crisp quantity $x' \in X$ and should be changed into a fuzzy set in order to use fuzzy inference rules. We usually define a singleton fuzzy set $A' \in \mathcal{F}(X)$ as

$$A'(x) = \begin{cases} 1, & x = x', \\ 0, & x \neq x'. \end{cases}$$

Since the output of the VWSI method is a fuzzy set $B' \in \mathcal{F}(Y)$, the centroid defuzzification method is often used to obtain the crisp output quantity in practice, i.e.,

$$y' = \frac{\int_Y yB'(y) dy}{\int_Y B'(y) dy}.$$

THEOREM 5.1. *Under the above conditions, there exists a group of base functions $\mathcal{A}' = \{A'_i \mid 1 \leq i \leq m\}$ such that the single-input single-output fuzzy system based on the VWSI algorithm defined by (9) and (13) is approximately a unary piecewise interpolation function*

$$f(x) = \sum_{i=1}^m A'_i(x)y_i, \quad (16)$$

and \mathcal{A}' is just a fuzzy partition of X .

PROOF. Let $h_1 = y_1 - c$, $h_i = y_i - y_{i-1}$ ($i = 2, \dots, m$) and $h = \max\{h_i \mid 1 \leq i \leq m\}$. Because \mathcal{A} and \mathcal{B} are fuzzy partitions, they have the Kronecker property, i.e., $A_i(x_j) = \delta_{ij} = B_i(y_j)$. For any input $x' \in X$, since $A'(x)$ is the singleton fuzzy set of it, $\sigma(A', A_i) = \bigvee_{x \in X} (A'(x) \wedge A_i(x)) = A_i(x')$ and the output quantity determined by (9) and (13) is

$$\begin{aligned} y' &= \frac{\int_c^d yB'(y) dy}{\int_c^d B'(y) dy} \approx \frac{\sum_{i=1}^m y_i B'(y_i) h_i}{\sum_{i=1}^m B'(y_i) h_i} = \frac{\sum_{i=1}^m y_i \left[\sum_{k=1}^m \frac{w_k^0 \cdot \sigma(A', A_k)}{\sum_{j=1}^m w_j^0 \cdot \sigma(A', A_j)} B_k(y_i) \right] h_i}{\sum_{i=1}^m \left[\sum_{k=1}^m \frac{w_k^0 \cdot \sigma(A', A_k)}{\sum_{j=1}^m w_j^0 \cdot \sigma(A', A_j)} B_k(y_i) \right] h_i} \\ &= \frac{\sum_{i=1}^m y_i \left[\sum_{k=1}^m \frac{w_k^0 A_k(x')}{\sum_{j=1}^m w_j^0 A_j(x')} B_k(y_i) \right] h_i}{\sum_{i=1}^m \left[\sum_{k=1}^m \frac{w_k^0 A_k(x')}{\sum_{j=1}^m w_j^0 A_j(x')} B_k(y_i) \right] h_i} = \frac{\sum_{i=1}^m y_i w_i^0 A_i(x') h_i}{\sum_{i=1}^m w_i^0 A_i(x') h_i}. \end{aligned}$$

Put $\alpha_i(x') \triangleq h_i / \sum_{j=1}^m w_j^0 A_j(x') h_j$ and $A'_i(x') \triangleq \alpha_i(x') w_i^0 A_i(x')$. Then

$$y' \approx \sum_{i=1}^m \alpha_i(x') (w_i^0 A_i(x')) y_i = \sum_{i=1}^m A'_i(x') y_i.$$

Let $\mathcal{A}' \triangleq \{A'_i \mid 1 \leq i \leq m\}$. It is easy to prove that \mathcal{A}' is a fuzzy partition of X . Put $f(x) \triangleq \sum_{i=1}^m A'_i(x) y_i$. Obviously, it is a unary piecewise interpolation function that takes A'_i ($1 \leq i \leq m$) as its base functions. Particularly, if $x' = x_i$, then $y' = y_i$. ■

NOTE 5.1. The proofs in the cases of fuzzy systems constructed by the other models and (13) are similar to that of Theorem 5.1.

THEOREM 5.2. *Under the assumption of Theorem 5.1, there exists a group of base functions $\phi = \{\phi_i \mid 1 \leq i \leq m\}$ such that the single-input single-output fuzzy system based on the VWSI algorithm defined by (9) and (14) is approximately a unary piecewise interpolation function*

$$f(x) = \sum_{i=1}^m \phi_i(x) y_i. \quad (17)$$

PROOF. For convenience, we suppose $\{x_i \mid 1 \leq i \leq m\}$ is an equidistant partition (i.e., $x_{i+1} - x_i = \lambda$ for all i). According to (14), we define $d(A', A_i) = |x' - x_i|/\lambda \wedge 1$. Then $0 \leq d(A', A_i) \leq 1$. For any $x' \in X$, there exists i_0 such that $x' \in [x_{i_0}, x_{i_0+1}]$. Then

$$S_{i_0} \triangleq S_{i_0}(d(A', A_1), \dots, d(A', A_m)) = 1 - \frac{|x' - x_{i_0}|}{\lambda},$$

$$S_{i_0+1} \triangleq S_{i_0+1}(d(A', A_1), \dots, d(A', A_m)) = 1 - \frac{|x' - x_{i_0+1}|}{\lambda}.$$

Meanwhile,

$$S_j(d(A', A_1), \dots, d(A', A_m)) = 1 - \frac{|x' - x_j|}{\lambda} \wedge 1 = 0,$$

where $j \neq i_0, i_0 + 1$. From (9) and (14), we can obtain

$$y' = \frac{\int_c^d y B'(y) dy}{\int_c^d B'(y) dy} \approx \frac{\sum_{i=1}^m y_i B'(y_i) \lambda}{\sum_{i=1}^m B'(y_i) \lambda} = \frac{w_{i_0}^0 S_{i_0} y_{i_0} + w_{i_0+1}^0 S_{i_0+1} y_{i_0+1}}{w_{i_0}^0 S_{i_0} + w_{i_0+1}^0 S_{i_0+1}}. \quad (18)$$

Let $\phi_i(x') = w_i^0 S_i / \sum_{j=1}^m w_j^0 S_j$ for any i , then $\phi_j(x') = 0$ ($j \neq i_0, i_0 + 1$). From (18), we have

$$y' = \sum_{i=i_0}^{i_0+1} \phi_i(x') y_i = \sum_{i=1}^m \phi_i(x') y_i.$$

Put $f(x) = \sum_{i=1}^m \phi_i(x) y_i$. Obviously, it is a unary piecewise interpolation function that takes ϕ_i ($1 \leq i \leq m$) as its base functions. Particularly, if $x' = x_i$, then $y' = y_i$. ■

NOTE 5.2. The proofs in the cases of fuzzy systems constructed by the other models and (14) are similar to that of Theorem 5.2.

LEMMA 5.1. Under the assumption of Theorem 5.1, if θ is a normal implication operator satisfying the following four conditions:

- (I.4) $\theta(a, a) = 1$,
- (I.5) $a \leq a' \Rightarrow \theta(a, b) \geq \theta(a', b)$,
- (I.6) $b \leq b' \Rightarrow \theta(a, b) \leq \theta(a, b')$,
- (I.7) $\theta(1, b) = b$,

then (15) becomes

$$w_i(A') = \frac{w_i^0 A_i(x')}{\sum_{j=1}^m w_j^0 A_j(x')}. \quad (19)$$

PROOF. Due to (I.4) and (I.5), when $a \leq b$, $\theta(a, b) \geq \theta(b, b) = 1$, so $\theta(a, b) = 1$. For any input quantity $x' \in X$, since $A'(x)$ is the singleton fuzzy set of it, we have

$$\begin{aligned} & \left(\bigwedge_{x \in X} \theta(A'(x), A_i(x)) \right) \vee \left(\bigwedge_{x \in X} \theta(A_i(x), A'(x)) \right) \\ &= \left[\theta(1, A_i(x')) \wedge \left(\bigwedge_{x \neq x'} \theta(0, A_i(x)) \right) \right] \vee \left[\theta(A_i(x'), 1) \wedge \left(\bigwedge_{x \neq x'} \theta(A_i(x), 0) \right) \right] \\ &= [\theta(1, A_i(x')) \wedge 1] \vee \left[1 \wedge \left(\bigwedge_{x \neq x'} \theta(A_i(x), 0) \right) \right] \\ &= \theta(1, A_i(x')) \vee \left(\bigwedge_{x \neq x'} \theta(A_i(x), 0) \right). \end{aligned}$$

When $x' = x_i$, the above expression is reduced to $\theta(1, A_i(x_i)) = A_i(x_i) = 1$. When $x' \neq x_i$, the above expression is reduced to $\theta(1, A_i(x')) \vee \theta(A_i(x_i), 0) = \theta(1, A_i(x')) \vee \theta(1, 0) = A_i(x')$. In any case, $(\bigwedge_{x \in X} \theta(A'(x), A_i(x))) \vee (\bigwedge_{x \in X} \theta(A_i(x), A'(x))) = A_i(x')$. Therefore, (15) changes into (19). ■

NOTE 5.3. Dubois and Prade proposed ten properties of fuzzy implication operators in [16], simplified as D-P properties. (I.4)–(I.7) in Lemma 5.1 are just (vi), (i), (ii) and (iv) in D-P properties, respectively.

Some familiar normal implication operators [13,17] satisfy conditions of Lemma 5.1, and include Wang implication operator

$$\theta(a, b) = \begin{cases} 1, & a \leq b, \\ (1 - a) \vee b, & a > b, \end{cases}$$

Lukasiewicz implication operator

$$\theta(a, b) = \begin{cases} 1, & a \leq b, \\ 1 - a + b, & a > b, \end{cases}$$

Goguen implication operator

$$\theta(a, b) = \begin{cases} 1, & a = 0, \\ \frac{b}{a} \wedge 1, & a > 0, \end{cases}$$

Gödel implication operator

$$\theta(a, b) = \begin{cases} 1, & a \leq b, \\ b, & a > b, \end{cases}$$

Dubois-Prade implication operator

$$\theta(a, b) = \begin{cases} (1 - a) \vee b, & (1 - a) \wedge b = 0, \\ 1, & \text{otherwise,} \end{cases}$$

$$\theta(a, b) = \begin{cases} 1, & a < 1, \\ b, & a = 1, \end{cases}$$

Modified Reichenbach implication operator

$$\theta(a, b) = \begin{cases} 1, & a \leq b, \\ 1 - a + ab, & a > b, \end{cases}$$

$$\theta(a, b) = \begin{cases} 1, & a \leq b, \\ (1 - a) \vee b \vee \frac{1}{2}, & 0 < b < a < 1, \\ (1 - a) \vee b, & \text{otherwise,} \end{cases}$$

$$\theta(a, b) = \begin{cases} 1 - a, & b = 0, \\ b, & a = 1, \\ 1, & \text{otherwise,} \end{cases}$$

and so on.

THEOREM 5.3. Under the assumption of Theorem 5.1, if θ is a normal implication operator satisfying (I.4)–(I.7) of Lemma 5.1, then there exists a group of base functions $\mathcal{A}' = \{A'_i \mid 1 \leq i \leq m\}$ such that the single-input single-output fuzzy system based on the VWSI algorithm defined by (9) and (15) is approximately a unary piecewise interpolation function

$$f(x) = \sum_{i=1}^m A'_i(x) y_i, \quad (20)$$

and \mathcal{A}' is just a fuzzy partition of X .

PROOF. Due to Lemma 5.1 and (15), we have

$$\begin{aligned}
 y' &= \frac{\int_c^d y B'(y) dy}{\int_c^d B'(y) dy} \approx \frac{\sum_{i=1}^m y_i B'(y_i) h_i}{\sum_{i=1}^m B'(y_i) h_i} = \frac{\sum_{i=1}^m y_i \left[\sum_{k=1}^m \frac{w_k(A')}{\sum_{j=1}^m w_j(A')} B_k(y_i) \right] h_i}{\sum_{i=1}^m \left[\sum_{k=1}^m \frac{w_k(A')}{\sum_{j=1}^m w_j(A')} B_k(y_i) \right] h_i} \\
 &= \frac{\sum_{i=1}^m y_i w_i(A') h_i}{\sum_{i=1}^m w_i(A') h_i} = \frac{\sum_{i=1}^m y_i w_i^0 A_i(x') h_i}{\sum_{i=1}^m w_i^0 A_i(x') h_i}.
 \end{aligned}$$

Denote $\alpha_i(x') \triangleq h_i / \sum_{j=1}^m h_j w_j^0 A_j(x')$ and $A'_i(x') \triangleq \alpha_i(x') w_i^0 A_i(x')$. Then

$$y' \approx \sum_{i=1}^m \alpha_i(x') w_i^0 A_i(x') y_i = \sum_{i=1}^m A'_i(x') y_i.$$

Put $\mathcal{A}' \triangleq \{A'_i \mid 1 \leq i \leq m\}$ and $f(x) \triangleq \sum_{i=1}^m A'_i(x) y_i$. It is easy to know that \mathcal{A}' is just a fuzzy partition of X . Particularly, if $x' = x_i$, then $y' = y_i$. \blacksquare

NOTE 5.4. The proofs in the cases of fuzzy systems constructed by the other models and (15) are similar to that of Theorem 5.3.

NOTE 5.5. From Theorems 5.1 and 5.3, the fuzzy system based on (9) and (13) is equivalent to the fuzzy system based on (9) and (15). In addition, though the Mamdani implication operator “ \wedge ” is not normal, (13) is a special case of (15). Hence, (I.3)–(I.7) are only sufficient conditions for (15) being a variable weight of rules such that Theorem 5.3 holds.

6. RELATIONSHIP BETWEEN VWSI SYSTEMS AND FAMILIAR FUZZY INFERENCE SYSTEMS

In this section, we prove that some other familiar fuzzy inference systems such as the Mamdani inference system, $(+, \cdot)$ -centroid inference system, simple inference system, function inference system and characteristic expansion inference system [1–6] are equivalent to certain special fuzzy systems constructed by VWSI method.

For convenience, we discuss multi-input single-output fuzzy systems. Some necessary concepts and notations for fuzzy systems are similar to those in Section 5.

$\prod_{i=1}^n k_i$ fuzzy inference rules are formed as follows:

If x_1 is A_{1j_1} and x_2 is A_{2j_2} and \dots and x_n is A_{nj_n} then y is $B_{j_1 j_2 \dots j_n}$,

where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$.

Generally speaking, the operation “ \wedge ” or “ \cdot ” is used to aggregate the antecedent components, i.e., for $x \triangleq (x_1, \dots, x_n)$, we put $A_{j_1 \dots j_n}(x) \triangleq \bigwedge_{i=1}^n A_{ij_i}(x_i)$ or $A_{j_1 \dots j_n}(x) \triangleq \prod_{i=1}^n A_{ij_i}(x_i)$.

THEOREM 6.1. *Under the assumption of Theorem 5.1, if $S_{j_1 \dots j_n} \triangleq \bigwedge_{i=1}^n A_{ij_i}(x_i)$ is taken and $w_{j_1 \dots j_n}^0$ are equal to each other, then the multi-input single-output fuzzy system based on the Mamdani algorithm is equivalent to the one based on the VWSI algorithm defined by (9) and (13), where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$.*

PROOF. According to the Mamdani algorithm, the inference relation of the $j_1 \dots j_n^{\text{th}}$ rule is $R_{j_1 \dots j_n} = A_{1j_1} \times \dots \times A_{nj_n} \times B_{j_1 \dots j_n}$, and the whole inference relation of $\prod_{i=1}^n k_i$ rules is

$$\begin{aligned}
 R &\triangleq \bigcup_{j_1=1}^{k_1} \dots \bigcup_{j_n=1}^{k_n} R_{j_1 \dots j_n}, \\
 R(x, y) &= \bigvee_{j_1=1}^{k_1} \dots \bigvee_{j_n=1}^{k_n} R_{j_1 \dots j_n}(x, y) = \bigvee_{j_1=1}^{k_1} \dots \bigvee_{j_n=1}^{k_n} \left(\bigwedge_{i=1}^n A_{ij_i}(x_i) \wedge B_{j_1 \dots j_n}(y) \right).
 \end{aligned}$$

For a given input $x' \triangleq (x'_1, \dots, x'_n)$, we have

$$B'(y) = \bigvee_{x \in X} \left(\bigwedge_{i=1}^n A'_i(x_i) \wedge R(x, y) \right) = \bigvee_{j_1=1}^{k_1} \cdots \bigvee_{j_n=1}^{k_n} \left(\bigwedge_{i=1}^n A_{ij_i}(x'_i) \wedge B_{j_1 \dots j_n}(y) \right).$$

The centroid defuzzification method is used to get the crisp output quantity

$$\begin{aligned} y' &= \frac{\int_Y y B'(y) dy}{\int_Y B'(y) dy} \approx \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} B'(y_{j_1 \dots j_n}) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} B'(y_{j_1 \dots j_n}) h_{j_1 \dots j_n}} \\ &= \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} \left(\bigvee_{i_1=1}^{k_1} \cdots \bigvee_{i_n=1}^{k_n} \left(\bigwedge_{p=1}^n A_{pi_p}(x'_p) \wedge B_{i_1 \dots i_n}(y_{j_1 \dots j_n}) \right) \right) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \left(\bigvee_{i_1=1}^{k_1} \cdots \bigvee_{i_n=1}^{k_n} \left(\bigwedge_{p=1}^n A_{pi_p}(x'_p) \wedge B_{i_1 \dots i_n}(y_{j_1 \dots j_n}) \right) \right) h_{j_1 \dots j_n}} \quad (21) \\ &= \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} \left(\bigwedge_{p=1}^n A_{pj_p}(x'_p) \right) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \left(\bigwedge_{p=1}^n A_{pj_p}(x'_p) \right) h_{j_1 \dots j_n}}. \end{aligned}$$

In accordance with the proof of Theorem 5.1, when $S_{j_1 \dots j_n} \triangleq \bigwedge_{i=1}^n A_{ij_i}(x_i)$ is taken and $w_{j_1 \dots j_n}^0$ are equal to each other, y' is just the crisp output quantity gotten by (9) and (13). ■

THEOREM 6.2. Under the assumption of Theorem 5.1, if $S_{j_1 \dots j_n} \triangleq \prod_{i=1}^n A_{ij_i}(x_i)$ is taken and $w_{j_1 \dots j_n}^0$ are equal to each other, then the multi-input single-output fuzzy system based on $(+, \cdot)$ -centroid algorithm is equivalent to the one based on the VWSI algorithm defined by (9) and (13), where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$.

PROOF. When (\vee, \wedge) in the Mamdani algorithm is replaced by $(+, \cdot)$, we obtain $(+, \cdot)$ -centroid algorithm. The whole inference relation of $\prod_{i=1}^n k_i$ rules is

$$R(x, y) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \prod_{p=1}^n A_{pj_p}(x_p) B_{j_1 \dots j_n}(y).$$

For a given input $x' \triangleq (x'_1, \dots, x'_n)$, we have $B'(y) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \prod_{p=1}^n A_{pj_p}(x'_p) B_{j_1 \dots j_n}(y)$. The centroid defuzzification method is used to get the crisp output quantity

$$\begin{aligned} y' &= \frac{\int_Y y B'(y) dy}{\int_Y B'(y) dy} \approx \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} B'(y_{j_1 \dots j_n}) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} B'(y_{j_1 \dots j_n}) h_{j_1 \dots j_n}} \\ &= \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} \left(\sum_{i_1=1}^{k_1} \cdots \sum_{i_n=1}^{k_n} \prod_{p=1}^n A_{pi_p}(x'_p) B_{i_1 \dots i_n}(y_{j_1 \dots j_n}) \right) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \left(\sum_{i_1=1}^{k_1} \cdots \sum_{i_n=1}^{k_n} \prod_{p=1}^n A_{pi_p}(x'_p) B_{i_1 \dots i_n}(y_{j_1 \dots j_n}) \right) h_{j_1 \dots j_n}} \quad (22) \\ &= \frac{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} y_{j_1 \dots j_n} \prod_{p=1}^n A_{pj_p}(x'_p) h_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \cdots \sum_{j_n=1}^{k_n} \prod_{p=1}^n A_{pj_p}(x'_p) h_{j_1 \dots j_n}}. \end{aligned}$$

In accordance with the proof of Theorem 5.1, when $S_{j_1 \dots j_n} \triangleq \prod_{i=1}^n A_{ij_i}(x_i)$ is taken and $w_{j_1 \dots j_n}^0$ are equal to each other, y' is just the crisp output quantity gotten by (9) and (13). ■

THEOREM 6.3. *Under the assumption of Theorem 5.1, if $S_{j_1 \dots j_n} \triangleq \prod_{i=1}^n A_{ij_i}(x_i)$ or $S_{j_1 \dots j_n} \triangleq \bigwedge_{i=1}^n A_{ij_i}(x_i)$ is taken, the output fuzzy partition is equidistant, and $w_{j_1 \dots j_n}^0$ are equal to each other, then the multi-input single-output fuzzy system based on simple inference algorithm is equivalent to the one based on the VWSI algorithm defined by (9) and (13), where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$.*

PROOF. In simple inference algorithm, rule consequent fuzzy sets are replaced by some real numbers, i.e., the inference rules are as follows:

If x_1 is A_{1j_1} and x_2 is A_{2j_2} and \dots and x_n is A_{nj_n} then y is $y_{j_1 \dots j_n}$,

where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$. For a given input $x' \triangleq (x'_1, \dots, x'_n)$, the output quantity y' is calculated in the following steps.

STEP 1. Every approach degree $\alpha_{j_1 \dots j_n}$ is defined by

$$\alpha_{j_1 \dots j_n}(x') = \prod_{p=1}^n A_{pj_p}(x'_p) \quad \text{or} \quad \alpha_{j_1 \dots j_n}(x') = \bigwedge_{p=1}^n A_{pj_p}(x'_p).$$

STEP 2.

$$y' = \frac{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} \alpha_{j_1 \dots j_n}(x') y_{j_1 \dots j_n}}{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} \alpha_{j_1 \dots j_n}(x')}. \quad (23)$$

According to the proof of Theorem 5.1, if $\{y_{j_1 \dots j_n} \mid 1 \leq j_i \leq k_i, 1 \leq i \leq n\}$ is an equidistant partition, $S_{j_1 \dots j_n} \triangleq \prod_{i=1}^n A_{ij_i}(x_i)$ or $S_{j_1 \dots j_n} \triangleq \bigwedge_{i=1}^n A_{ij_i}(x_i)$ is taken, and $w_{j_1 \dots j_n}^0$ are equal to each other, then y' is just the crisp output quantity gotten by (9) and (13). ■

THEOREM 6.4. *Under the assumption of Theorem 5.1, if the approach degree operator $g_{j_1 \dots j_n}$ in function inference algorithm is regarded as $S_{j_1 \dots j_n}$, the output fuzzy partition is equidistant, and $w_{j_1 \dots j_n}^0$ are equal to each other, then the multi-input single-output fuzzy system based on function inference algorithm is equivalent to the one based on the VWSI algorithm defined by (9) and (13), where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$.*

PROOF. The function inference method was proposed in [5], which is a generalization of the simple inference method. The inference rules are as follows:

If x_1 is A_{1j_1} and x_2 is A_{2j_2} and \dots and x_n is A_{nj_n} then y is $y_{j_1 \dots j_n}(x_1, \dots, x_n)$,

where $j_i = 1, \dots, k_i$ and $i = 1, \dots, n$. For a given input $x' \triangleq (x'_1, \dots, x'_n)$, the approach degree is defined by $g_{j_1 \dots j_n}$, i.e., $\alpha_{j_1 \dots j_n}(x') \triangleq g_{j_1 \dots j_n}(A_{1j_1}(x'_1), \dots, A_{nj_n}(x'_n))$, where $g_{j_1 \dots j_n}$ is a general operator, for example, $g_{j_1 \dots j_n} = \wedge$ or $g_{j_1 \dots j_n} = \cdot$. Then

$$\begin{aligned} y' &= \frac{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} \alpha_{j_1 \dots j_n}(x') y_{j_1 \dots j_n}(x')}{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} \alpha_{j_1 \dots j_n}(x')} \\ &= \frac{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} g_{j_1 \dots j_n}(A_{1j_1}(x'_1), \dots, A_{nj_n}(x'_n)) y_{j_1 \dots j_n}(x'_1, \dots, x'_n)}{\sum_{j_1=1}^{k_1} \dots \sum_{j_n=1}^{k_n} g_{j_1 \dots j_n}(A_{1j_1}(x'_1), \dots, A_{nj_n}(x'_n))}. \end{aligned} \quad (24)$$

According to the proof of Theorem 5.1, if $\{y_{j_1 \dots j_n} \mid 1 \leq j_i \leq k_i, 1 \leq i \leq n\}$ is an equidistant partition, $g_{j_1 \dots j_n}$ is regarded as $S_{j_1 \dots j_n}$, $w_{j_1 \dots j_n}^0$ are equal to each other and put $y_{j_1 \dots j_n} \triangleq y_{j_1 \dots j_n}(x_1, \dots, x_n)$, then the crisp output quantity is the y' gotten by (24). ■

NOTE 6.1. In [6], it was pointed out that the characteristic expansion inference algorithm is equivalent to the Mamdani algorithm. Therefore, the fuzzy system constructed by characteristic expansion inference algorithm is equivalent to the one based on the VWSI algorithm defined by (9) and (13).

7. A METHOD FOR DETERMINING CONSTANT WEIGHTS OF RULES AND A SIMULATION EXPERIMENT

7.1. A Method for Determining Constant Weights of Fuzzy Rules

We use the approach for extracting fuzzy rules from database, proposed in [18], to compare VWSI systems with traditional Mamdani inference systems. Specially, the constant weight of every rule in VWSI systems can be explained as a fuzzy conditional probability that the rule consequent appears when the rule antecedent happens. The concept of fuzzy probability was introduced by Zadeh in [19]. Let (X, \mathcal{F}, P) be a probability space. A fuzzy set A on X is called a fuzzy event if its membership function $A(x)$ is measurable on (X, \mathcal{F}) . The fuzzy probability of A is defined by

$$P(A) = \int_X A(x) dP.$$

On the basis of the given data set $D = \{(x_j^0, y_j^0) \mid 1 \leq j \leq q\}$ (i.e., D contains q pairs of known data), we give the estimate expression of the fuzzy conditional probability as a constant weight of the i^{th} rule

$$w_i^0 = \frac{\sum_{j=1}^q A_i(x_j^0) B_i(y_j^0)}{\sum_{k=1}^m \sum_{j=1}^q A_i(x_j^0) B_k(y_j^0)}. \quad (25)$$

Since $\{B_i \mid 1 \leq i \leq m\}$ is a fuzzy partition of Y , we have $\sum_{k=1}^m B_k(y) = 1$ for all $y \in Y$. Then (25) can be reduced to

$$w_i^0 = \frac{\sum_{j=1}^q A_i(x_j^0) B_i(y_j^0)}{\sum_{j=1}^q A_i(x_j^0) \sum_{k=1}^m B_k(y_j^0)} = \frac{\sum_{j=1}^q A_i(x_j^0) B_i(y_j^0)}{\sum_{j=1}^q A_i(x_j^0)}. \quad (26)$$

7.2. VWSI Systems in Time Series Prediction

The prediction of time series appears to be an important practical issue, and can be applied to many fields such as economics, control, signal processing, and so on. We use a VWSI system to predict Mackey-Glass chaotic time series generated by the differential equation with time delay

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t). \quad (27)$$

When $\tau > 17$, the graph of (27) shows a chaotic action. So we choose $\tau = 30$ [20] (see Figure 1).

The prediction of time series utilizes known time series data at moment t to predict the value at a future moment $t + P$. In general, a mapping from q sampling points $(x(t - (q-1)I), \dots, x(t - I), x(t))$ to $x(t + P)$ is constructed, where I is a time interval unit. For instance, we may choose $q = 4$, $I = P = 6$. Then for every moment t , the input of the fuzzy system is a four-dimensional vector $X(t) = (x(t-18), x(t-12), x(t-6), x(t))$, and the output is the predicted value $y(t) = x(t+6)$. Accordingly, we generate 1000 input-output data as the integer moment t

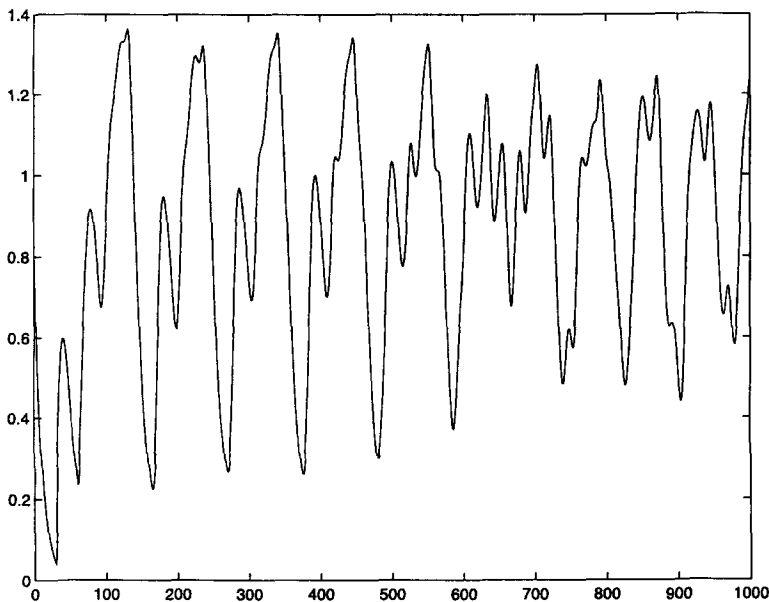


Figure 1. Partial graph of Mackey-Glass chaotic time series.

varies from 118 to 1117. The first 500 data are used to generate fuzzy rules with constant weights and construct a VWSI system, and the rest 500 data are used to test the approximation effect of the new inference system.

We use triangular membership functions, commonly used in practice, for rule antecedents (refer to Figure 2). Their membership functions have the forms

$$A_1(x) = \begin{cases} \frac{x - x_2}{x_1 - x_2}, & x_1 \leq x \leq x_2, \\ 0, & \text{otherwise,} \end{cases}$$
$$A_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x \leq x_i, \\ \frac{x - x_{i+1}}{x_i - x_{i+1}}, & x_i \leq x \leq x_{i+1}, \\ 0, & \text{otherwise,} \end{cases}$$
$$(i = 2, \dots, m - 1)$$

$$A_m(x) = \begin{cases} \frac{x - x_{m-1}}{x_m - x_{m-1}}, & x_{m-1} \leq x \leq x_m, \\ 0, & \text{otherwise,} \end{cases}$$

where x_i is the peak point of fuzzy set A_i ($1 \leq i \leq m$).

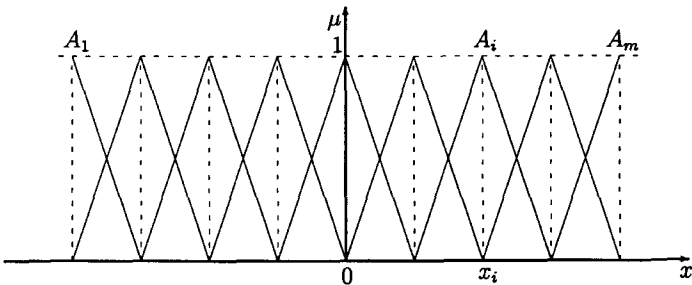


Figure 2. Membership functions of antecedent fuzzy sets.

It should be noted that there are four input variables in this experiment. Generally, those four fuzzy inputs are aggregated by the operator “.” or “ \wedge ”, i.e., $A_i(x') \triangleq \prod_{k=1}^4 A_{ik}(x'_{ik})$ or $A_i(x') \triangleq \bigwedge_{k=1}^4 A_{ik}(x'_{ik})$. For the convenience of computing, we use the former.

Figure 3 gives the mean square errors (MSE) produced by the new system and the traditional Mamdani inference system approximating a part of Mackey-Glass chaotic time series according to the different numbers of fuzzy rules. It shows that the approximation degree of those two systems is improved as the number of fuzzy rules increases. Nevertheless, for the same number of fuzzy rules, the approximation degree of the VWSI system with constant weights of rules appears to be better than that of the traditional Mamdani inference system.

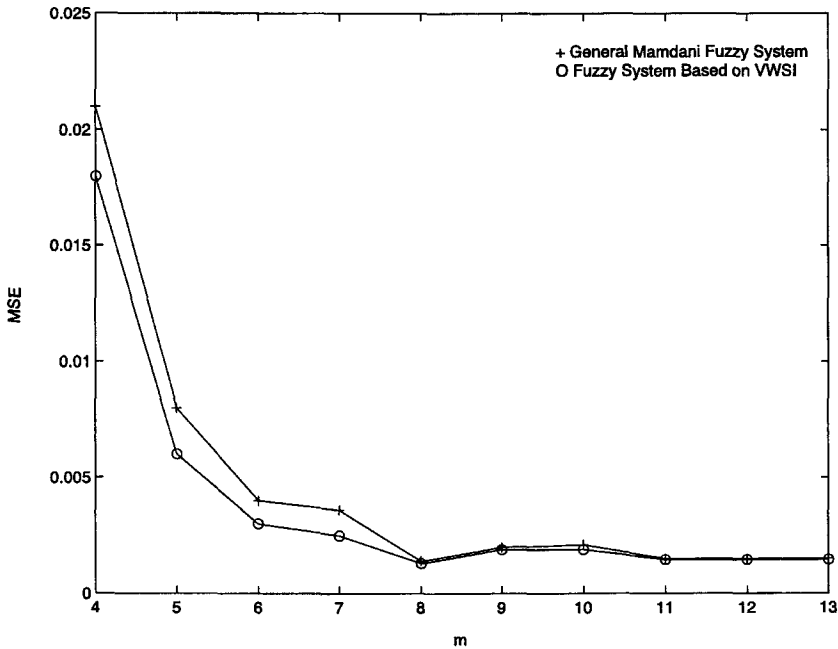


Figure 3. Comparison between mean square errors of two systems.

8. CONCLUSIONS

The VWSI method was built on the basis of the principle of variable weighted synthesis in the factor spaces theory. In addition to the consideration of the relative credibility of each rule, the weight values of inference rules can be adjusted dynamically in accordance with different input quantities, and regarded as degrees of activation of inference rules. The entire inference process of VWSI method conforms to human thought, and the method can lay a theoretical foundation for studying the adaptivity mechanism of fuzzy controllers. Through analyzing the relationship between VWSI systems and other familiar fuzzy inference systems, we could conclude that VWSI method is extensively meaningful.

Reference [9] pointed out that gradient vectors of balance functions are state variable weight vectors, and provided a general approach to construct state variable weight vectors. We only chose some simple forms of state variable weights in the present paper. We can construct more extensive state variable weights of inference rules in terms of balance functions, and then may further enrich VWSI algorithms.

REFERENCES

1. L.A. Zadeh, Fuzzy sets, *Information and Control* **8**, 338–353, (1965).
2. M. Mizumoto, Original fuzzy control method, (in Japanese), *Science of Mathematical Physics* **333**, 27–35, (1991).

3. T. Terano, K. Asai and M. Sugeno, *Fuzzy Systems Theory and its Applications*, Academic Press, Boston, MA, (1992).
4. M. Sugeno, *Fuzzy Control*, Japanese Industry News Press, Tokyo, (1988).
5. T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics* **15** (1), 116–132, (1985).
6. Y.Y. Chen and T.Y. Chen, Characteristic expansion inference method, (in Chinese), *J. Liaoning Normal Univ.* **3**, 1–8, (1984).
7. G.J. Wang, Full implication triple I method for fuzzy reasoning, (in Chinese), *Science in China, (Series E)* **29** (1), 43–53, (1999).
8. P.Z. Wang, *Fuzzy Sets and Falling Shadows of Random Sets*, (in Chinese), Beijing Normal University Press, Beijing, (1985).
9. H.X. Li, Factor spaces and mathematical frame of knowledge representation (VIII), (in Chinese), *Fuzzy Systems and Mathematics* **9** (3), 1–9, (1995).
10. H.X. Li, Multifactorial functions in fuzzy sets theory, *Fuzzy Sets and Systems* **35** (1), 69–84, (1990).
11. W.X. Zhang and G.X. Liang, *Fuzzy Control and Systems*, (in Chinese), Xi'an Jiaotong University Press, Xi'an, (1998).
12. H.X. Li, *Fuzzy Mathematics*, (in Chinese), National Defense Industry Press, Beijing, (1994).
13. W.M. Wu, *Principle and Methods of Fuzzy Reasoning*, (in Chinese), Guizhou Science and Technology Press, Guiyang, (1994).
14. H.X. Li, Interpolation mechanism of fuzzy control, *Science in China (Series E)* **41** (3), 312–320, (1998).
15. H.X. Li, F. You and J.Y. Peng, Fuzzy controllers based on some fuzzy implication operators and their response functions, (in Chinese), *Progress in Natural Science* **14** (1), 15–20, (2004).
16. D. Dubois and H. Prade, Fuzzy sets in approximate reasoning, *Fuzzy Sets and Systems* **40** (1), 143–244, (1991).
17. G.J. Wang, *Non-Classical Mathematical Logic and Approximate Reasoning*, (in Chinese), Science Press, Beijing, (2000).
18. L.X. Wang and J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man and Cybernetics* **22** (6), 1414–1427, (1992).
19. L.A. Zadeh, Probability measures of fuzzy events, *J. Math. Anal. Appl.* **23**, 421–427, (1968).
20. L.X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, (in Chinese), National Defense Industry Press, Beijing, (1995).
21. H.X. Li, L.D. Xu, J.Y. Wang and Z.W. Mo, Feature space theory in data mining: Transformations between extensions and intensions in knowledge representation, *Expert Systems* **20** (2), 60–71, (2003).
22. H.X. Li and L.D. Xu, Feature space theory—A mathematical foundation for data mining, *Knowledge-Based Systems* **14** (5/6), 253–257, (2001).
23. S. Feng and L.D. Xu, Decision support for fuzzy comprehensive evaluation of urban development, *Fuzzy Sets and Systems* **105**, 1–12, (1999).